HIVE: HISTORY-INITIALIZED VISUOMOTOR EVOLUTION WITH A COMPACT VISUAL BOTTLENECK FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models jointly predict future actions and visual outcomes, but noise-initialized generation can require costly iterative refinement. While some policies initialize generation from semantically informative priors, they predict actions alone without jointly forecasting visual outcomes. We present HIVE (History-Initialized Visuomotor Evolution), a flow-matching framework that uses historical actions and observed visual outcomes as the starting state for joint future prediction. This structured prior starts closer to the target than Gaussian noise, enabling more accurate generation with fewer integration steps while retaining the predictive visual supervision of world-action models. To keep joint modeling efficient, we introduce an Action-Centric Visual Bottleneck (ACVB), a lightweight 28.5M-parameter compressor that reduces 196 frozen DINOv3 patch tokens to very few compact, manipulation-informative tokens per frame. Dense-feature and policy supervision preserve local interaction information in a shared representation for history and future targets. Causal temporal attention captures action–outcome dependencies. Across 18 RoboTwin tasks, six-step HIVE improves mean success by 12.06 percentage points over its action-only prediction variant, and by 13.84 and 9.34 points over six- and ten-step noise-initialized variants, respectively. Compared with the ten-step variant, it reduces mean inference latency by 56.3% per eight-action chunk. Using ACVB instead of global CLS features improves mean success by 17.89 points over HIVE-CLS. It achieves 71% mean success on five real-world tasks, supporting effective and efficient history-initialized visuomotor prediction.

1 INTRODUCTION

World-action models (WAMs) couple action generation with future visual prediction, learning how a scene evolves during robotic interaction (Li et al., 2025; Kim et al., 2026; Li et al., 2026). Many initialize the generated variables from noise and use observations only as conditions. Yet executed actions and their observed outcomes provide a structured record of ongoing interaction grounded in recent physical experience. An attempted grasp, for example, may lift an object or leave it in place; the action alone does not reveal which outcome occurred. Pairing actions with visual responses exposes this distinction and the state from which subsequent behavior must proceed. Can this history itself serve as the starting state for generating future actions together with their visual outcomes?

Action-to-Action flow matching (A2A) (Jia et al., 2026) initializes action generation from historical actions under visual conditioning, exploiting temporal continuity to improve generation with fewer inference steps, including high-quality single-step prediction. Its robot policy predicts actions alone. Joint visual prediction adds supervision for the environment's response, encouraging the learned dynamics to connect robot behavior with scene changes and anticipate the consequences of subsequent actions. We extend this principle to a joint source and target: historical actions paired with their observed visual outcomes are transported toward future actions and visual states (Fig. 1). A2A's frames-to-frames extension also improves future-frame prediction over a matched regression baseline, providing empirical precedent for history-initialized visual prediction (Jia et al., 2026, Sec. 6).

Joint prediction requires visual features that are informative yet compact. CLS tokens and average pooling can discard local interaction details that are essential for precise grasping and manipula-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

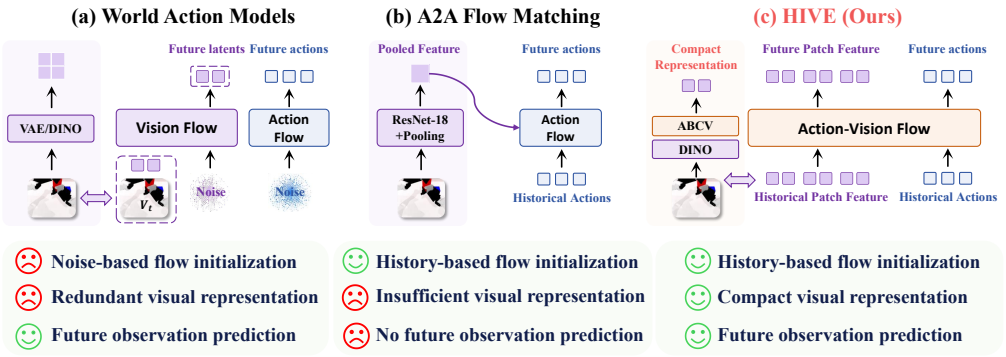


Figure 1: Comparison of generation strategies. (a) World-action models predict future actions and visual states from noise, often using many visual tokens with redundant information. (b) A2A initializes action generation from historical actions, conditioned on pooled visual features with limited spatial detail. (c) HIVE initializes joint visual–action generation from historical actions and compact, manipulation-relevant visual tokens produced by ACVB.

tion (Zhou et al., 2026; Qian et al., 2024), whereas reconstruction-oriented visual latents and dense patch sequences carry substantial information and computational cost beyond what control may require (Li et al., 2025; Lyu et al., 2026). Representing both history and future makes this token budget especially consequential. The desired representation should preserve object–gripper relations and local changes within a compact representation of both historical and future frames.

We propose **HIVE** (**H**istory-**I**nitialized **V**isuomotor **E**volution), a flow-matching framework for joint visual–action generation from interaction history. Its **Action-Centric Visual Bottleneck (ACVB)** compresses frozen DINOv3 patch features (Siméoni et al., 2025) into a small set of learned tokens per frame. Dense-feature supervision and policy objectives train these tokens to retain manipulation-relevant information, providing a shared representation for historical inputs and future targets without image reconstruction. HIVE interleaves each action with its subsequent visual observation and transports this sequence toward the joint future. Causal temporal attention preserves action–outcome ordering while all future positions are refined in parallel, without autoregressive rollout. History provides a starting point closer to the future target, reducing the refinement needed for accurate generation. In our experiments, six-step HIVE outperforms noise initialization even with ten steps, while reducing inference latency (Sec. 5.4).

We evaluate HIVE on 18 RoboTwin tasks with third-person single-arm Franka and egocentric dual-arm ARX-X5 robots, and five real-world tasks. Ablations examine visual prediction, history initialization, visual compression, and causal interaction. Our contributions are:

History-initialized joint generation. We introduce HIVE, which transports historical actions and their observed visual outcomes toward joint future trajectories, enabling effective and efficient generation. Causal attention preserves temporal ordering between interleaved modalities.

Compact, task-adaptive visual representation. We develop ACVB to compress dense pretrained features into a small token set shared by history and future prediction. Dense-feature and policy supervision retain local interaction information without image reconstruction.

Evaluation across robot configurations. We evaluate HIVE on 18 simulated and five real-world tasks spanning single-arm and bimanual manipulation. Controlled ablations examine visual supervision, history initialization, token compression, and causal attention across both robot configurations.

2 Related Work

2.1 World-Action Models for Robotic Manipulation

Beyond action-focused visuomotor imitation (Florence et al., 2022; Shafiullah et al., 2022; Zhao et al., 2023; Wang et al., 2023; Ze et al., 2024; Liang et al., 2026), world-action models use future visual prediction to support control (Kim et al., 2026; Li et al., 2026; Ye et al., 2026; Hu et al., 2024). Video-

based planners recover actions through inverse dynamics (Du et al., 2023; Zhou et al., 2024), while joint models predict actions alongside observations (Wu et al., 2024; Cen et al., 2025; Cheang et al., 2024; Xu et al., 2026). UVA (Li et al., 2025) predicts future observations and actions from historical observations and actions, whereas UWM (Zhu et al., 2025) couples video and action diffusion for pretraining on robotic datasets; other approaches exploit training-time video supervision (Yuan et al., 2026) or unify policy, dynamics, and visual forecasting (Lyu et al., 2026). HIVE initializes joint generation from observed interactions to predict future actions and visual outcomes.

2.2 Informed Initialization for Generative Policies

Diffusion Policy (Chi et al., 2025) and π_0 (Black et al., 2024) generate actions from noise using diffusion and flow matching, respectively. Informed initialization replaces noise with structured inputs: VITA (Gao et al., 2026) transports visual representations into action latents, while A2A (Jia et al., 2026) initializes from historical proprioceptive sequences under visual conditioning. HIVE extends this idea to joint visual–action transport, using both historical actions and observed outcomes as the source and predicting future behavior together with the resulting visual changes.

2.3 Visual Representations and Token Compression for Robot Control

Visual pretraining supplies language–image (Radford et al., 2021; Zhai et al., 2023), video-predictive (Bardes et al., 2024), and dense image priors (Siméoni et al., 2025) for control. Patch Policy (Zhou et al., 2026) and SOFT (Qian et al., 2024) report limitations of CLS tokens and global average pooling for spatially precise control. They retain patch-level or object-centric information, while DVK (Chang et al., 2025) extracts semantic keypoints from DINO patches to improve generalization to unseen objects; TokenLearner (Ryoo et al., 2021) and RT-1 (Brohan et al., 2022) aggregate spatial features, while VLA-Pruner (Liu et al., 2025) selects tokens by semantic and action relevance. ACVB uses dense-feature and policy supervision to learn shared compact DINOv3 tokens for both historical observations and future visual targets.

3 Preliminary

Notation and flow matching. Let V_t denote the visual representation of I_t, observed before executing a_t; the resulting observation is I_{t+1}. We use H historical steps to predict the next F steps. Following the notation in Method, Z^F denotes a future target representation. Given a source Z^S of the same shape and optional conditioning information $\mathcal{C}$, flow matching (Lipman et al., 2022) defines an interpolated state and its target velocity as

$$Z_\tau = (1-\tau)Z^S + \tau Z^F, \qquad V_\tau^* = Z^F - Z^S, \tag{1}$$

where $\tau \sim \mathcal{U}(0,1)$ is flow time, distinct from interaction time t. The training objective is

$$\mathcal{L}_{FM} = \mathbb{E}\Big[\big\| f_\theta(Z_\tau, \tau, \mathcal{C}) - (Z^F - Z^S)\big\|_2^2\Big], \tag{2}$$

with expectation over training pairs and flow time. Inference integrates $dZ_\tau/d\tau = f_\theta(Z_\tau, \tau, \mathcal{C})$ from $Z_0 = Z^S$ to obtain the predicted terminal state $\hat{Z}^F$.

Noise-initialized world-action prediction. WAMs connect action prediction with future visual prediction (Li et al., 2025; Lyu et al., 2026). UVA conditions future video and action prediction on observed interaction history, using separate diffusion heads. The following shared-flow formulation abstracts noise-initialized joint prediction rather than reproducing a specific model architecture. Let Z_A^{gt} and Z_V^{gt} denote the encoded future action and visual targets. The joint state is generated from noise under historical conditioning:

$$Z^S = \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad Z^F = (Z_A^{gt}, Z_V^{gt}), \quad \mathcal{C} = H_t. \tag{3}$$

where $H_t = \{(a_i, V_{i+1})\}_{i=t-H}^{t-1}$ is the available interaction history, and the pair denotes a joint state without prescribing its token ordering. Both interpolated modalities enter the predictor and receive velocity updates; historical inputs condition the flow from the Gaussian source, with interpolation $Z_\tau = (1-\tau)\epsilon + \tau Z^F$ between noise and the joint future target.

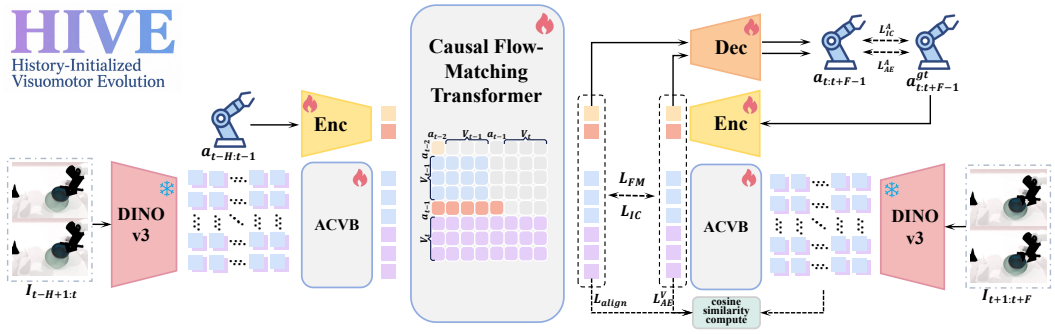


Figure 2: Overview of HIVE, illustrated with $H = F = 2$ for clarity (experiments use $H = F = 8$ unless otherwise specified). Frozen DINOv3 features are compressed by ACVB and interleaved with action embeddings. The causal flow-matching Transformer transports the historical sequence toward joint future visual–action latents. Action reconstruction, visual alignment, and inference consistency objectives supervise the shared representation and its decoded actions.

Action-to-action flow matching. A2A (Jia et al., 2026) replaces noise with encoded historical actions and transports them toward future action latents, conditioned on observed visual features:

$$Z^S = Z_A^H = E_a(A_{t-H:t-1}), \quad Z^F = Z_A^{gt} = E_a(A^{gt}_{t:t+F-1}), \quad \mathcal{C} = C_t^V = E_{\text{vis}}(I_{\le t}). \tag{4}$$

where E_{vis} aggregates the available historical images into a global visual condition. This condition modulates the velocity network through adaptive layer normalization (AdaLN), also used in robotic diffusion Transformers (Dasari et al., 2025), in its AdaLN-MLP blocks. Thus, the transported state contains action latents, while vision guides the flow without being predicted. The terminal action latent is decoded as $\hat{A}_{t:t+F-1} = D_a(\hat{Z}_A^F)$. Our method instead sets $Z^S = Z^H$ to the joint visual–action history and retains a joint future target Z^F, without an additional conditioning input, so observed interactions directly initialize the generated trajectory.

4 Method

4.1 Overview

HIVE transports interaction history toward future actions and visual states (Fig. 2). At time t, the robot observes I_t, executes a_t, and receives I_{t+1}. We denote a single action by a_i, an action chunk by $A_{i:j} = [a_i, \dots, a_j]$, and the history and future horizons by H and F. With action embeddings $z_i^a = E_a(a_i)$ and visual token blocks Z_i^V representing I_i, the source and target sequences are

$$X^H = [z^a_{t-H}, Z^V_{t-H+1}, \dots, z^a_{t-1}, Z^V_t], \qquad X^F = [z^a_t, Z^V_{t+1}, \dots, z^a_{t+F-1}, Z^V_{t+F}]. \tag{5}$$

Each action precedes its observed outcome: history ends at I_t, and future prediction begins with a_t and I_{t+1}. We use $H = F = 8$, giving source and target matching shapes and transition ordering.

4.2 Action-Centric Visual Bottleneck

Frozen DINOv3 (Siméoni et al., 2025) provides pretrained visual priors and native tokens compatible with our Transformer. We first represent each frame by its global CLS token, yielding HIVE-CLS (Appendix A). However, Sec. 5.5 shows that this frozen global representation has weaker correspondence with manipulation-relevant regions and lower downstream task success.

To retain richer local information with few tokens, we introduce ACVB, a trainable Transformer that maps each image's $P = 196$ patch features to M compact tokens, using $M = 3$ in our experiments: The token-count choice is supported by the visual and task-level ablation in Appendix G.

$$P_i = E^{patch}_{\text{DINO}}(I_i) \in \mathbb{R}^{P\times 768}, \qquad Z_i^V = E_{\text{ACVB}}(P_i) \in \mathbb{R}^{M\times 768}. \tag{6}$$

A shared ACVB encodes history and future targets. Dense-feature and policy supervision preserve local information and adapt the tokens to manipulation without pixel reconstruction.

4.3 History-Aware Flow Matching

The action encoder maps each $a_i \in \mathbb{R}^{D_a}$ to the visual token dimension. Let $Z_H = X^H$ and $Z_F = X^F$; each contains $H(1+M)$ tokens when $H = F$. HIVE instantiates the flow in Eqs. 1–2 with source $Z^S = Z_H$, and target $Z^F = Z_F$. The velocity field $f_\theta(Z_\tau, \tau)$ uses history in the transported state, with no language or separate condition. It learns the velocity $Z_F - Z_H$ along the history-to-future interpolation between paired visual–action sequences of matching length.

Causal attention over the interpolated sequence permits preceding transitions and interaction within each visual block, but prevents an action token from accessing its subsequent observation. This preserves temporal action–outcome dependencies while refining all future positions in parallel. At inference, ODE integration from $Z_0 = Z_H$ yields the terminal state $\hat{Z}_F$. Its action tokens are decoded as $\hat{a}_{t+i} = D_a(\hat{z}^a_{t+i})$ for $i = 0, \ldots, F-1$; its visual blocks predict $I_{t+1:t+F}$ in feature space.

4.4 Training Objectives

We jointly train the encoders, flow Transformer, and action decoder with the combined objective:

$$\begin{aligned}\mathcal{L}_{HIVE} = \lambda_{FM}(\mathcal{L}^A_{FM} + \mathcal{L}^V_{FM}) + \lambda^A_{AE}\mathcal{L}^A_{AE} + \lambda^V_{AE}\mathcal{L}^V_{AE} \\ + \lambda_{IC}(\mathcal{L}^A_{IC} + \mathcal{L}^V_{IC}) + \lambda_{align}\mathcal{L}_{align}.\end{aligned} \tag{7}$$

Here $\mathcal{L}^A_{FM}$ and $\mathcal{L}^V_{FM}$ are the action and visual components of Eq. 2. Action reconstruction $\mathcal{L}^A_{AE}$ applies an L_1 loss to $D_a(E_a(A^{gt}))$ and $A^{gt} = A^{gt}_{t:t+F-1}$.

For visual supervision, each frozen target patch is matched to its most cosine-similar compact token. The mean mismatch supervises both predicted visual tokens ($\mathcal{L}_{align}$) and target-side ACVB outputs ($\mathcal{L}^V_{AE}$). The latter constrains compression to preserve dense feature information, while allowing token specialization; it does not require exact patch or image reconstruction. Both branches share ACVB, with gradients updating the shared compressor while the DINOv3 backbone remains frozen.

Following A2A (Jia et al., 2026), inference consistency supervises the ODE-integrated endpoint: $\mathcal{L}^A_{IC}$ aligns inferred actions with ground truth in both latent and decoded action spaces, while $\mathcal{L}^V_{IC}$ aligns inferred visual tokens with compact ACVB targets. Appendix F gives the complete loss definitions.

5 Experiments

5.1 Experimental Setup

We evaluate 18 RoboTwin (Chen et al., 2025) tasks: 12 single-arm Franka tasks with a third-person camera and six distinct bimanual ARX-X5 tasks with an egocentric camera. For each Franka task, we use 50 clean demonstrations and 100 evaluation trials, varying scene configurations between training and testing. Five real-world tasks cover dual-arm ARX A7 and single-arm Franka robots. Methods share training data, epochs, observation/action horizons, and evaluation protocols unless specified; experiments use one RTX 4090 GPU (24 GB).

Baselines span diffusion (DP), noise-initialized flow (FM-N6/N10), and history-initialized action generation (A2A), alongside VITA, Patch Policy, and HIVE-CLS. FM-N6/N10 use A2A's backbone and action encoder, with historical action and visual features as adaLN conditions; 6/10 denotes training-time consistency integration steps. Ablations isolate visual supervision, initialization, representation, and temporal attention. Appendix B reports their detailed per-task results.

5.2 Main Results

Table 1 compares eight policies across 18 tasks. HIVE achieves 57.06% mean success, exceeding HIVE-CLS (39.17%) and the strongest non-HIVE baseline, A2A (35.33%), by 17.89 and 21.72 percentage points. HIVE ranks first on 16 tasks and second on the other two, showing gains across both robot–camera configurations rather than a small subset of tasks.

The advantage spans generation paradigms: DP reaches 9.89%, while noise-initialized FM-N6/N10 reach 27.44%/31.61%. Increasing consistency integration steps improves the latter baseline but

Table 1: Success rates (%) on 18 RoboTwin tasks. Bold and underline mark the best and second-best values per row, including ties. Mean averages all 18 tasks.

Task	DP	VITA	Patch Policy	FM-N6	FM-N10	A2A	HIVE-CLS	HIVE
Franka: third-person								
Adjust Bottle	44	72	61	90	86	78	83	**93**
Beat Block Hammer	0	4	46	0	0	21	35	**75**
Click Alarmclock	44	38	58	46	46	55	58	**92**
Click Bell	37	18	42	53	56	45	57	**89**
Open Laptop	10	37	0	36	53	51	58	**61**
Place Shoe	0	0	1	3	2	0	10	**15**
Place Empty Cup	1	0	0	1	0	2	5	**23**
Move Playing Card Away	2	18	18	10	18	23	22	**47**
Press Stapler	9	49	25	49	41	41	21	**71**
Turn Switch	13	36	49	25	31	25	29	**51**
Rotate QR Code	0	29	0	20	17	34	32	**40**
Place Object Stand	0	6	8	21	24	**31**	15	28
ARX-X5: egocentric								
Grab Roller	3	53	72	19	42	66	68	**98**
Handover Block	0	0	18	5	2	4	21	**25**
Place Container Plate	7	38	53	67	52	63	67	**71**
Place Burger Fries	0	30	24	13	18	23	39	**45**
Move Can Pot	8	29	**39**	29	29	18	35	38
Stack Bowls Two	0	62	54	7	52	56	50	**65**
Mean	9.89	28.83	31.56	27.44	31.61	35.33	39.17	**57.06**

remains below HIVE. Patch Policy (Zhou et al., 2026) also compresses DINOv3 patch-level features, but achieves 31.56% mean success versus 57.06% for HIVE across the 18 tasks. The HIVE–HIVE-CLS gap further supports using adaptable spatial tokens within joint prediction. These comparisons motivate the controlled ablations below, which separate visual supervision, initialization, representation, and temporal attention.

The lightweight HIVE-small variant achieves 52.11% with 22.69M trainable parameters, 76.44% fewer than HIVE and 9.17M fewer than VITA, the next-smallest model with reported counts. On the six ARX-X5 tasks its 55.17% mean remains close to HIVE's 57.00%. Detailed results and parameter comparisons appear in Appendices D and E. Appendix E.1 also reports latency and memory: HIVE-small lowers peak allocated memory by 36.2% with DINOv3, with slightly lower inference latency.

5.3 Does Visual Supervision Improve Action Generation?

HIVE-VS and **HIVE-CLS-VS** retain their parent architectures and historical visual inputs but predict only action velocities. They remove visual flow, consistency, and alignment losses, including target-side ACVB reconstruction, while retaining all action objectives. Visual supervision raises 18-task mean success from 45.00% to 57.06% for HIVE and from 28.89% to 39.17% for HIVE-CLS (Fig. 3).

Using checkpoints from the same training epoch, we measure OOD joint-action MAE on four Franka and three ARX-X5 tasks (Fig. 4). Visual supervision reduces error on all seven tasks for HIVE and six for HIVE-CLS. This supports future visual supervision as a useful training signal for control under domain shift.

5.4 Why Does History Initialization Improve Flow Matching?

HIVE-N6/N10 initialize from Gaussian noise and use historical actions and visual tokens as conditions through causal cross-attention. Their consistency objectives integrate six/ten steps during training; HIVE uses six here. At the same six-step budget, HIVE reaches 57.06% mean success versus 43.22% for HIVE-N6, a 13.84-point gain (Fig. 5). Increasing the noise baseline to ten steps raises success to 47.72%, still 9.34 points below six-step HIVE. HIVE also lowers mean full-inference latency from 73.57 to 32.16 ms per eight-action chunk, including DINOv3 (Appendix E.1). Thus, history initialization achieves better control with fewer steps and lower latency than HIVE-N10.

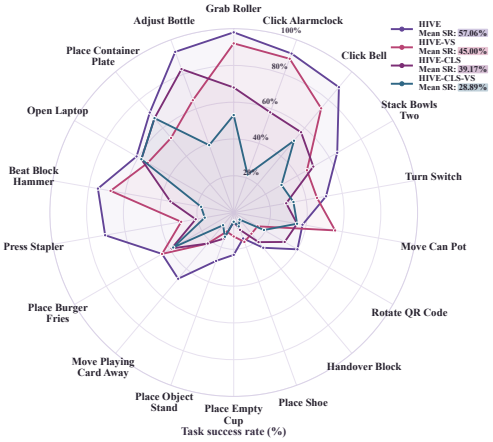


Figure 3: Visual-supervision ablation across all 18 RoboTwin tasks. HIVE-VS and HIVE-CLS-VS remove visual supervision from HIVE and HIVE-CLS, respectively.

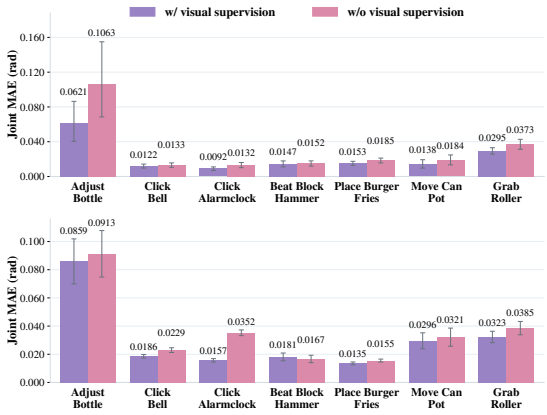


Figure 4: OOD joint-action MAE on seven tasks. Top: HIVE versus HIVE-VS. Bottom: HIVE-CLS versus HIVE-CLS-VS. The first four tasks use Franka and the the last three use ARX-X5.

For OOD inputs, we track all-token RMSE $d_k = \| \mathrm{vec}(Z_{\tau_k} - Z_F)\|_2/\sqrt{D}$, where D counts scalar latent entries and $\tau_k = k/N$ for N Euler steps. Figure 6 shows two representative Franka tasks, averaged over input sequences. History starts closer to the target and ends with lower error: HIVE reaches 0.0153/0.0231 on the two tasks, versus 0.0555/0.0678 for HIVE-N6 and 0.0586/0.0676 for HIVE-N10. Observed interactions thus provide a more informative source for future transport.

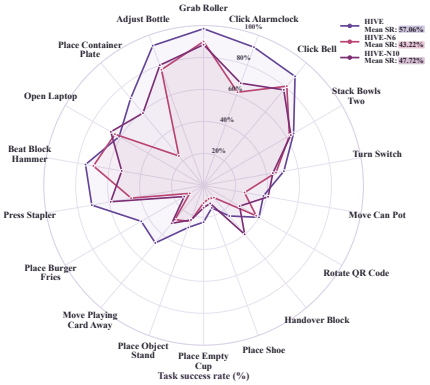


Figure 5: History versus noise initialization across 18 tasks. HIVE-N6 and HIVE-N10 use Gaussian initialization with six and ten consistency integration steps, respectively.

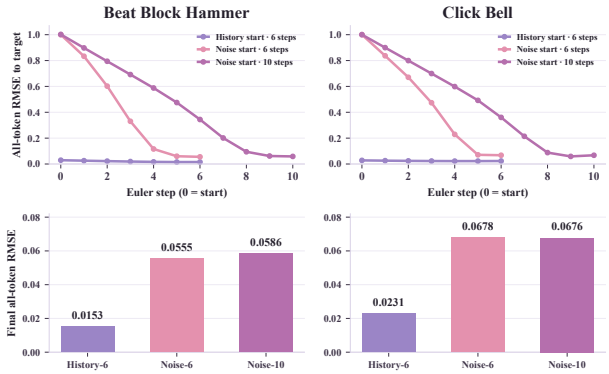


Figure 6: OOD flow analysis on Beat Block Hammer and Click Bell. For each task, the upper panel shows all-token RMSE to the future target at each Euler step, The lower panel shows terminal RMSE. History-6 denotes HIVE; Noise-6 and Noise-10 denote HIVE-N6 and HIVE-N10.

5.5 Why Do We Need the Action-Centric Visual Bottleneck?

Figure 7 compares CLS and ACVB tokens against 196 DINOv3 patches using cosine similarity on OOD images from two Franka and two ARX-X5 tasks. It shows the final three history frames. The maps reveal spatial feature correspondences. The frozen CLS maps (blue) are weaker or more diffuse over manipulation-relevant regions; ACVB tokens (pink) exhibit complementary correspondence with objects, robot parts, and background without fixed semantic assignments. ACVB uses three tokens per frame, about $65\times$ fewer than the dense patch grid, with 28.5M trainable parameters.

Replacing A2A's trainable ResNet18 with frozen CLS features, projected to the same 512 dimensions, lowers mean success from 35.33% to 24.00% (Fig. 8). This comparison changes both the backbone

and its adaptability. Within the same HIVE framework, ACVB improves mean success from HIVE-CLS's 39.17% to 57.06% (Table 1), supporting compact, task-adaptive spatial features.

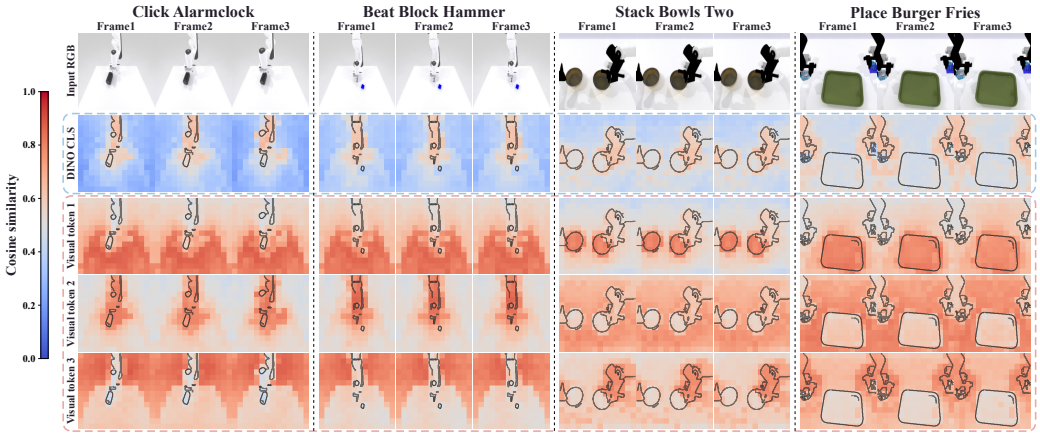


Figure 7: Cosine similarity to DINOv3 patch features on OOD images from four tasks. Each task shows history frames 6–8. Blue outlines mark CLS maps; pink outlines group the three learned ACVB-token maps. The tokens exhibit complementary correspondence with objects, robot parts, and background across adjacent observations.

Linear action probes. We fit separate ridge probes per task using frozen visual features: CLS, the mean of three ACVB tokens (ACVB-Mean), or their concatenation (ACVB-Concat). Each predicts either one action from one frame or eight future actions from eight history frames. Table 2 reports training-set fits, macro-averaged over 18 task–robot combinations (protocol and task coverage in Appendix H). At equal feature dimension, ACVB-Mean lowers MSE relative to CLS by 7.56% and 8.03% in the two settings. ACVB-Concat improves all three metrics further, but triples probe input dimensionality. These results support greater linear action decodability of the task-adapted features.

Table 2: Visual ridge probes: training-set fits averaged over 18 task–robot combinations. Bold/underline mark best/second-best results. CLS and ACVB-Mean have equal dimensions.

Single frame → single action				Eight history frames → eight future actions			
Feature	MSE ↓	MAE ↓	R^2 ↑	Feature	MSE ↓	MAE ↓	R^2 ↑
CLS	0.05832	0.13071	0.79918	CLS	0.06193	0.13479	0.79949
ACVB-Mean	<u>0.05392</u>	<u>0.12545</u>	<u>0.81705</u>	ACVB-Mean	<u>0.05696</u>	<u>0.12797</u>	<u>0.82308</u>
ACVB-Concat	**0.05117**	**0.12132**	**0.82442**	ACVB-Concat	**0.05361**	**0.12336**	**0.83163**

5.6 Does Causal Temporal Interaction Matter?

HIVE-Bi and **HIVE-CLS-Bi** replace the causal mask with full bidirectional attention while retaining other settings. Mean success falls from 57.06% to 46.83% for HIVE and from 39.17% to 34.61% for HIVE-CLS (Fig. 9). The consistent gains support preserving temporal action–outcome ordering in joint generation, with benefits extending across both visual representation choices.

5.7 Real-World Experiments

We evaluate three egocentric dual-arm ARX A7 tasks (Pick Two Cube, Drawer Cube Place, Grab Stick) and two third-person Franka tasks (Click Bell, Beat Block Hammer), shown in Fig. 10. Object placements vary across trials; Appendix C shows example configurations. HIVE ranks first on all five tasks and averages 71% success, compared with 57% for noise-initialized HIVE-N, 49% for HIVE-VS, 44% for HIVE-CLS, and 32% for A2A (Table 3). These results extend the simulation findings to physical manipulation across both robot–camera configurations.

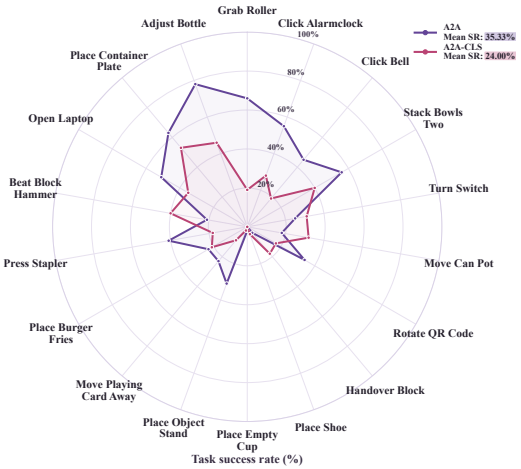


Figure 8: A2A versus A2A-CLS across all 18 RoboTwin tasks. Both provide a 512-dimensional visual representation to the policy.

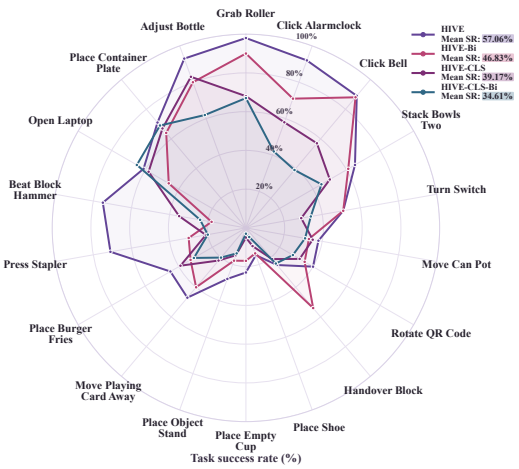


Figure 9: Causal versus bidirectional attention on 18 tasks. HIVE-Bi and HIVE-CLS-Bi use full bidirectional attention.

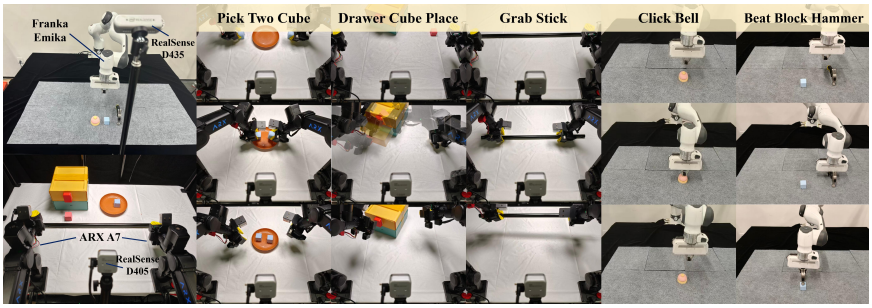


Figure 10: Real-world setups and tasks. Left: Franka with a third-person camera and dual-arm ARX A7 with an egocentric camera. Right: demonstrations of the three bimanual tasks and two single-arm tasks used for evaluation.

Table 3: Real-world success rates (%). The first three tasks use ARX A7; the last two use Franka. Each task is evaluated over 20 trials. Bold indicates the best result in each column.

	ARX A7 (egocentric)			Franka (third-person)		
Method	Pick Two Cube	Drawer Cube Place	Grab Stick	Click Bell	Beat Block Hammer	Mean
A2A	15	25	40	50	30	32
HIVE-CLS	30	50	45	60	35	44
HIVE-VS	40	45	50	65	45	49
HIVE-N	50	55	60	70	50	57
HIVE	**65**	**70**	**75**	**80**	**65**	**71**

6 Conclusion and Future Work

HIVE initializes joint visual–action flow matching from interaction history, combining compact ACVB tokens with causal temporal attention. It achieves 57.06% mean success on 18 RoboTwin tasks and 71% on five real-world tasks, outperforming noise initialization with fewer steps and lower latency. Future work will explore more advanced flow-matching backbones, incorporate language conditioning, and scale pretraining to build an efficient and capable robot foundation model. These extensions aim to improve generalization across tasks and embodiments, support language-guided manipulation, and transfer the benefits of history initialization to broader robotic interaction settings.

AI USE STATEMENT

We used generative AI tools to assist with several aspects of this work. Specifically, AI was used to (i) polish and improve the clarity of the manuscript writing, (ii) help search for and retrieve related literature, (iii) assist in revising and refining figures and diagrams in the paper, and (iv) support the analysis and interpretation of experimental results. All research ideas, experimental design, implementation, and final scientific conclusions were solely the responsibility of the authors. The authors carefully reviewed, verified, and take full responsibility for the entire content of this paper.

REPRODUCIBILITY STATEMENT

To support reproduction, Appendix D details the network architectures, visual preprocessing, and parameter counts; Appendix F specifies the training objectives and loss weights; and Appendix H provides the visual probing protocol. RoboTwin in-domain data collection starts from the default random seed 0, while out-of-domain test data collection starts from random seed 100.

REFERENCES

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Wei-Di Chang, Francois Hogan, Scott Fujimoto, David Meger, and Gregory Dudek. Generalizable imitation learning through pre-trained representations. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE, 2025.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15617–15625. IEEE, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pp. 158–168. PMLR, 2022.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy. In *International Conference on Learning Representations*, volume 2026, pp. 100903–100933, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. *arXiv preprint arXiv:2602.07322*, 2026.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model. *arXiv preprint arXiv:2503.00200*, 2025. URL https://arxiv.org/abs/2503.00200.

Qiwei Liang, Boyang Cai, Minghao Lai, Sitong Zhuang, Tao Lin, Yan Qin, Yixuan Ye, Jiaming Liang, and Renjing Xu. Bootstrap dynamic-aware 3d visual representation for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13419–13429, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. *arXiv preprint arXiv:2210.02747*, 2022. URL https://arxiv.org/abs/2210.02747.

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Vla-pruner: Temporal-aware dual-level visual token pruning for efficient vision-language-action inference. *arXiv e-prints*, pp. arXiv–2511, 2025.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Jianing Qian, Anastasios Panagopoulos, and Dinesh Jayaraman. Recasting generic pretrained vision transformers as object-centric scene encoders for manipulation policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 17544–17552. IEEE, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*, 2021. URL https://arxiv.org/abs/2103.00020.

Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Xiaoxu Xu, Hao Li, Jinhui Ye, Yilun Chen, Jia Zeng, Xinyi Chen, Linning Xu, Dahua Lin, Weixin Li, and Jiangmiao Pang. Futurevla: Joint visuomotor prediction for vision-language-action model. *arXiv preprint arXiv:2603.10712*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. *arXiv preprint arXiv:2403.03954*, 2024. URL https://arxiv.org/abs/2403.03954.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. *arXiv preprint arXiv:2303.15343*, 2023. URL https://arxiv.org/abs/2303.15343.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Gaoyue Zhou, Zichen Jeff Cui, Ada Langford, Bowen Tan, Yann LeCun, and Lerrel Pinto. Patch policy: Efficient embodied control via dense visual representations. *arXiv preprint arXiv:2607.18236*, 2026.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

APPENDIX

A HIVE-CLS: GLOBAL VISUAL REPRESENTATION VARIANT

HIVE-CLS is a controlled variant of HIVE that replaces the learned ACVB token block with a single frozen DINOv3 CLS token per observation. It retains history initialization, causal temporal attention, and the action encoder and decoder described in Sec. 4. Figure 11 illustrates this variant.

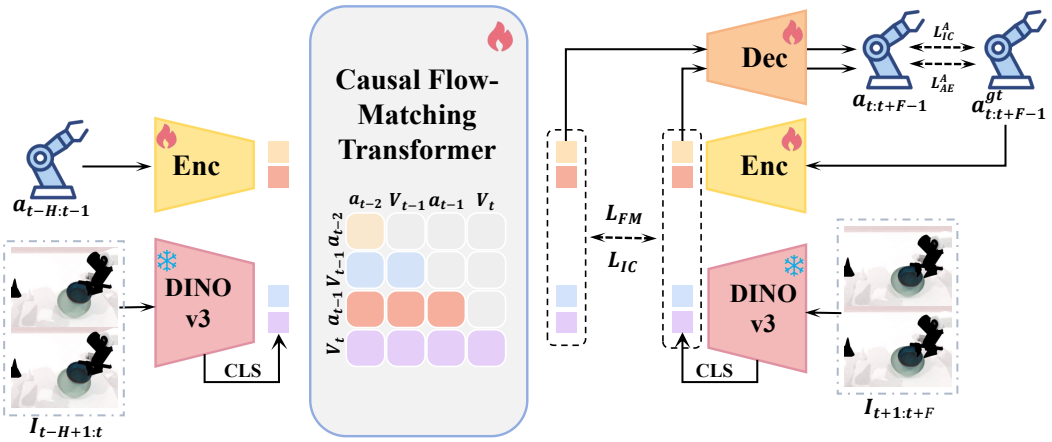


Figure 11: HIVE-CLS architecture, illustrated with $H = F = 2$. Frozen DINOv3 CLS tokens are interleaved with action embeddings and transported from history to future by causal flow matching. The supervision branch supplies future CLS targets for $I_{t+1:t+F}$, together with future actions.

For each image I_i, the visual representation is

$$c_i = E^{CLS}_{\text{DINO}}(I_i) \in \mathbb{R}^{768}. \tag{8}$$

The historical and future latent sequences become

$$X^H = [z^a_{t-H}, c_{t-H+1}, \ldots, z^a_{t-1}, c_t], \qquad X^F = [z^a_t, c_{t+1}, \ldots, z^a_{t+F-1}, c_{t+F}]. \tag{9}$$

They contain $2H$ and $2F$ tokens, respectively, with $H = F$. Each action token precedes the observation resulting from its execution. The causal mask lets each token attend to itself and preceding positions: c_{i+1} can attend to z^a_i, while z^a_i cannot attend to c_{i+1}. The terminal flow state contains predicted action and CLS tokens, and D_a decodes the action tokens as in HIVE.

The frozen visual targets and visual inference consistency loss are

$$Z^{gt}_V = E^{CLS}_{\text{DINO}}(I_{t+1:t+F}), \qquad \mathcal{L}^V_{IC} = \left| \hat{Z}^V_F - Z^{gt}_V \right|_1. \tag{10}$$

Since this variant uses frozen CLS features without ACVB, it omits both prediction-side dense-feature alignment and target-side visual reconstruction. The action objectives are unchanged, and its total loss is

$$\mathcal{L}_{CLS} = \lambda_{FM}(\mathcal{L}^A_{FM} + \mathcal{L}^V_{FM}) + \lambda^A_{AE}\mathcal{L}^A_{AE} + \lambda_{IC}(\mathcal{L}^A_{IC} + \mathcal{L}^V_{IC}). \tag{11}$$

The visual flow-matching term uses the frozen CLS targets. Comparisons with HIVE in Sec. 5.5 assess the effect of replacing compact, trainable spatial features with this global representation.

B DETAILED RESULTS FOR ADDITIONAL POLICIES

Table 4 complements Table 1 with the policies used in the radar comparisons but not listed in the main table. All entries are success rates in percent. Means are unweighted averages over the indicated tasks. The suffixes -VS and -Bi denote removal of visual supervision and use of bidirectional attention, respectively; HIVE-N6/N10 use Gaussian initialization with six/ten consistency steps.

Table 4: Per-task success rates (%) for additional policies on RoboTwin. Franka and ARX-X5 means use 12 and 6 tasks, respectively; the overall mean uses all 18 tasks.

Task	HIVE-N6	HIVE-N10	HIVE-VS	HIVE-Bi	A2A-CLS	HIVE-CLS-VS	HIVE-CLS-Bi
Franka (third-person)							
Adjust Bottle	77	80	65	80	46	39	62
Beat Block Hammer	70	52	68	18	40	18	24
Click Alarmclock	62	68	89	71	28	22	42
Click Bell	81	78	74	88	19	51	39
Open Laptop	64	67	54	46	35	58	65
Place Shoe	9	12	17	14	4	8	5
Place Empty Cup	11	14	13	17	0	5	3
Move Playing Card Away	28	31	22	40	9	9	20
Press Stapler	46	59	29	30	18	16	20
Turn Switch	46	44	46	51	31	33	34
Rotate QR Code	38	26	15	35	17	19	28
Place Object Stand	24	23	11	18	2	14	14
Franka mean	46.33	46.17	41.92	42.33	20.75	24.33	29.67
ARX-X5 (egocentric)							
Grab Roller	90	88	92	90	19	53	67
Handover Block	10	40	15	54	18	5	24
Place Container Plate	24	59	53	64	53	67	69
Place Burger Fries	10	14	45	33	21	38	31
Move Can Pot	26	41	56	33	32	35	31
Stack Bowls Two	62	63	46	61	40	30	45
ARX-X5 mean	37.00	50.83	51.17	55.83	30.50	38.00	44.50
Overall mean	43.22	47.72	45.00	46.83	24.00	28.89	34.61

C Real-World Initial Object Configurations

The real-world evaluation varies the initial object placements across trials of each task. Figure 12 presents five sampled configurations for each of the five tasks. These examples document variation in object position and relative arrangement, so the reported success rates assess robustness across initial spatial configurations.

D Model Architectures and Parameter Counts

Table 5 specifies HIVE, HIVE-small, and HIVE-CLS for the 14-dimensional bimanual action configuration. All use $H = F = 8$ and a frozen DINOv3 ViT-B/16 backbone with 85,660,416 parameters. Images are resized to 224×224 and normalized. DINOv3 produces 196 patch tokens of dimension 768; its CLS and four register tokens are excluded from ACVB. HIVE-CLS instead retains only the global CLS token.

ACVB processes each frame independently with learned patch-position embeddings, pre-LayerNorm residual attention/FFN blocks, GELU, and dropout 0.1. It selects the final three patch positions and applies LayerNorm; there are no additional query tokens or temporal attention within ACVB. HIVE-small projects these 768-dimensional features into its 256-dimensional flow space and maps predicted visual features back to 768 dimensions for feature supervision. The flow blocks use per-head query/key normalization, RoPE, and six-way time modulation. A 256-dimensional sinusoidal time embedding is mapped through a 1024-dimensional Mish MLP to the flow width.

History and future branches share ACVB and the action encoder; action reconstruction and flow prediction share the decoder. Table 6 counts each shared module once. HIVE-CLS removes ACVB

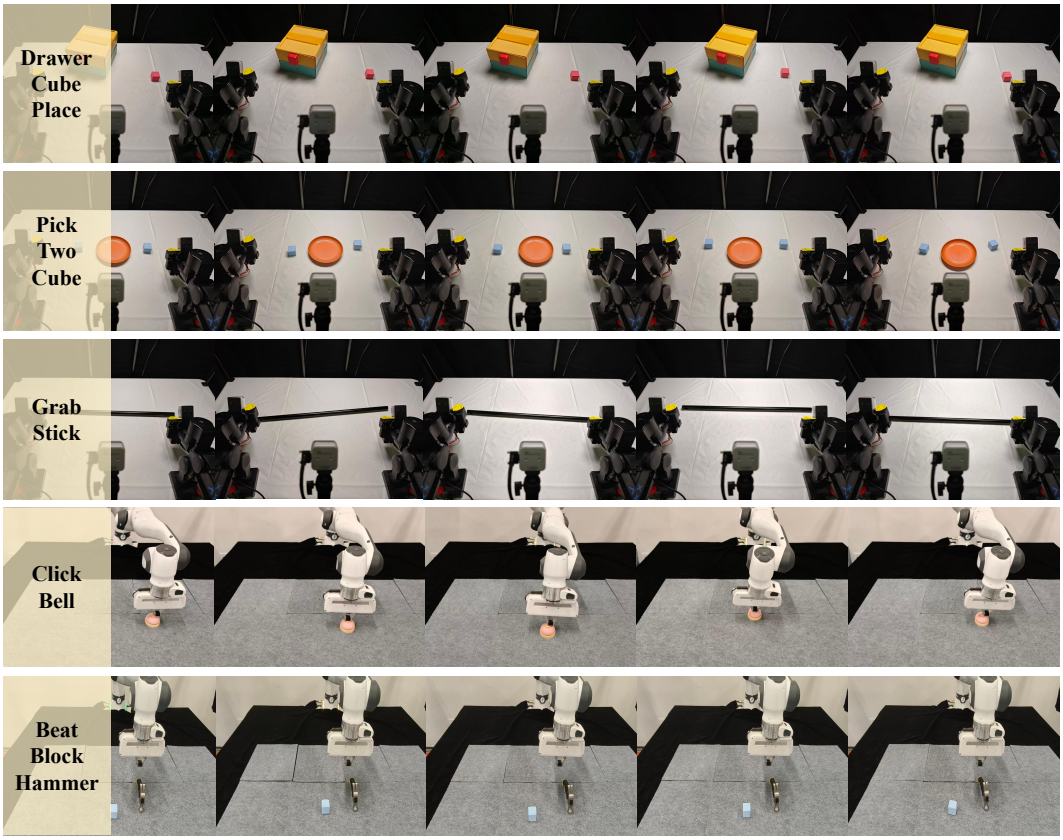


Figure 12: Examples of initial object configurations used in real-world evaluation. Each row shows five sampled arrangements for one task. From top to bottom: Drawer Cube Place, Pick Two Cube, Grab Stick, Click Bell, and Beat Block Hammer. The first three tasks use dual-arm ARX A7; the last two use Franka.

Table 5: Architectures of the three HIVE variants. Dimensions and action heads correspond to the 14-dimensional configuration.

Component	HIVE	HIVE-small	HIVE-CLS
Frozen visual backbone	DINOv3 ViT-B/16	DINOv3 ViT-B/16	DINOv3 ViT-B/16
ACVB layers / width	4 / 768	3 / 768	–
ACVB heads / FFN width	6 / 3072	6 / 1536	–
Visual tokens per frame	3	3	1 (CLS)
Visual input projection	Identity	$768 \to 256$	Identity
Action encoder	$14 \to 768 \to 768$	$14 \to 256 \to 256$	$14 \to 768 \to 768$
Joint sequence shape	32×768	32×256	16×768
Flow layers / width / heads	6 / 768 / 6	6 / 256 / 4	6 / 768 / 6
Flow FFN width	3072	1024	3072
Action decoder	$768 \to 768 \to 768 \to 14$	$256 \to 256 \to 256 \to 14$	$768 \to 768 \to 768 \to 14$
Visual supervision projection	Identity	$256 \to 768$	Identity

while retaining HIVE's remaining trainable modules. Frozen backbone parameters are excluded from the trainable totals but remain part of each deployed model.

Table 6: Exact trainable parameter counts by module. The frozen DINOv3 backbone (85.66M parameters) is excluded.

Module	HIVE	HIVE-small	HIVE-CLS
ACVB	28,503,552	14,333,184	0
Visual input projection	0	196,864	0
Action encoder (state MLP)	602,112	69,632	602,112
Modality embeddings	1,536	512	1,536
Flow Transformer	66,006,272	7,760,128	66,006,272
Action decoder	1,191,950	135,182	1,191,950
Visual supervision projection	0	197,376	0
Total	96,305,422	22,692,878	67,801,870

E Success Rates and Trainable Model Size

Table 7 compares trainable model size and task success, with separate columns for the same 12-task Franka and six-task ARX-X5 subsets. HIVE-small reduces trainable parameters from 96.31M to 22.69M, a 76.44% reduction, while retaining 52.11% mean success versus 57.06% for HIVE. The decrease is 6.50 percentage points on Franka and 1.83 points on ARX-X5. On ARX-X5, it improves Handover Block and ties Place Container Plate; the largest decrease is on Place Burger Fries (30% versus 45%). Thus, the close bimanual average does not imply uniform performance across tasks.

Table 7: Trainable parameters and mean success rates (%). M denotes millions of parameters; frozen backbone parameters are excluded for HIVE variants. – indicates unavailable data. Success rates are compared within each task subset.

Method	Trainable parameters (M)	Franka 12 tasks	ARX-X5 6 tasks	Overall 18 tasks
DP	96.89	13.33	3.00	9.89
VITA	31.86	25.58	35.33	28.83
Patch Policy	32.33	25.67	43.33	31.56
A2A	34.75	33.83	38.33	35.33
HIVE-CLS	67.80	35.42	46.67	39.17
HIVE	96.31	57.08	57.00	57.06
HIVE-small	**22.69**	**50.58**	**55.17**	**52.11**

Parameter accounting. HIVE counts refer to the 14-dimensional configuration in Table 6. Baseline counts are reported implementation values. A2A's 34.75M includes its trainable 11.18M-parameter ResNet18 visual encoder.

HIVE-small uses fewer trainable parameters and exceeds the reported Franka success rates of DP, VITA, Patch Policy, A2A, and HIVE-CLS. Compared with HIVE-CLS, it uses fewer trainable parameters and achieves higher success on both robot configurations. These results support a favorable parameter–success trade-off.

Table 8: Per-task success rates (%) for HIVE and HIVE-small. Means are unweighted over the indicated tasks.

Task	HIVE	HIVE-small
Franka (third-person)		
Adjust Bottle	93	75
Beat Block Hammer	75	67
Click Alarmclock	92	85
Click Bell	89	85
Open Laptop	61	72
Place Shoe	15	13
Place Empty Cup	23	18
Move Playing Card Away	47	18
Press Stapler	71	65
Turn Switch	51	51
Rotate QR Code	40	30
Place Object Stand	28	28
Franka mean	57.08	50.58
ARX-X5 (egocentric)		
Grab Roller	98	96
Handover Block	25	35
Place Container Plate	71	71
Place Burger Fries	45	30
Move Can Pot	38	36
Stack Bowls Two	65	63
ARX-X5 mean	57.00	55.17
Overall mean	57.06	52.11

E.1 Inference Latency and GPU Memory

Table 9 reports the supplementary median timings for HIVE-small and HIVE under FP32, batch size one, eight input frames, eight output actions, and six flow integration steps. Latency is per eight-action chunk. Full inference includes DINOv3 but excludes camera acquisition, environment rendering, and robot execution. Components are timed independently and need not sum to the full-inference time.

Table 9: HIVE-small versus HIVE. Latencies are supplementary-run medians (ms per eight-action chunk); peak allocated GPU memory (MiB) comes from the original profiling run. Lower is better.

Metric	HIVE-small	HIVE	Reduction
Full inference, including DINOv3	30.44	31.66	3.8%
Visual compressor	1.81	3.23	43.8%
Flow integration, ordinary calls	22.80	23.46	2.8%
Flow integration, CUDA Graph replay	2.84	5.45	47.9%
Peak allocated memory, including DINOv3	495.05	775.90	36.2%

HIVE-small substantially reduces compressor latency and peak memory, while its full-inference latency is only modestly lower. CUDA Graph replay is an integration diagnostic, not end-to-end deployment latency; its larger reduction indicates that execution overhead affects the realized speed benefit. The table retains peak allocated memory rather than mixing it with reserved memory or device-wide occupancy. HIVE-N10 uses ten steps and lacks a matching supplementary median measurement, so it is excluded from this six-step comparison.

History versus noise initialization. In the original profiling run, HIVE (six steps) and HIVE-N10 (ten steps) require 32.16 and 73.57 ms mean full-inference latency per eight-action chunk, respectively, including DINOv3. This is a 56.3% latency reduction with 40% fewer integration steps. These measurements use FP32, batch size one, and eight input frames; they exclude camera acquisition, environment rendering, and robot execution. Both numbers are means from the same run,

distinct from the supplementary medians in Table 9. The comparison measures the complete policies, including their different history-conditioning mechanisms.

F Detailed Training Objectives

This section expands the objective in Eq. 7. We set $\lambda_{AE}^{V} = \lambda_{align} = 0.1$ and $\lambda_{FM} = \lambda_{AE}^{A} = \lambda_{IC} = 1.0$ in all experiments. All losses are averaged over training examples. We write $A^{gt} = A^{gt}_{t:t+F-1}$ and obtain $\hat{Z}_F$ by integrating the learned flow from Z_H over $\tau \in [0, 1]$. Superscripts A and V select the action and visual positions of this terminal sequence.

Action reconstruction. The same action encoder processes historical and future actions, and the shared decoder reconstructs encoded actions and decodes predictions:

$$\mathcal{L}_{AE}^{A} = \left\| D_a(E_a(A^{gt})) - A^{gt} \right\|_1 . \tag{12}$$

Visual feature alignment and target reconstruction. For compact tokens $U = [u_1, \ldots, u_M]$ and frozen dense patch features $P_j^{gt} = [p_{j,1}^{gt}, \ldots, p_{j,P}^{gt}]$, define

$$\ell_{align}(U, P_j^{gt}) = \frac{1}{P} \sum_{p=1}^{P} \left(1 - \max_{1 \le r \le M} \frac{u_r^{\top} p_{j,p}^{gt}}{\|u_r\|_2 \|p_{j,p}^{gt}\|_2} \right). \tag{13}$$

Each patch is matched to its most similar compact token, encouraging the token set to cover dense visual features without prescribed regions. Let $Z_j^{\star} = E_{\mathrm{ACVB}}(P_j^{gt})$ and collect the future visual target as $Z_V^{\star} = [Z_{t+1}^{\star}, \ldots, Z_{t+F}^{\star}]$. These blocks form the visual positions of Z_F, interleaved with actions at $t, \ldots, t+F-1$. The prediction-side and target-side losses are

$$\mathcal{L}_{align} = \frac{1}{F} \sum_{j=t+1}^{t+F} \ell_{align}(\hat{Z}_j^{V}, P_j^{gt}), \qquad \mathcal{L}_{AE}^{V} = \frac{1}{F} \sum_{j=t+1}^{t+F} \ell_{align}(Z_j^{\star}, P_j^{gt}). \tag{14}$$

Here $\hat{Z}_j^{V}$ is the predicted block in $\hat{Z}_F$ for observation I_j. The second term directly supervises the three target-side ACVB tokens against all 196 source patches. Gradients update the shared ACVB while DINOv3 remains frozen. This reconstruction-style feature constraint reduces information loss without a patch decoder or pixel reconstruction.

Inference consistency. Following A2A (Jia et al., 2026), the action endpoint is supervised in both latent and original action spaces, with $\lambda_A > 0$:

$$\mathcal{L}_{IC}^{A} = \left\| \hat{Z}_F^{A} - E_a(A^{gt}) \right\|_1 + \lambda_A \left\| D_a(\hat{Z}_F^{A}) - A^{gt} \right\|_1, \qquad \mathcal{L}_{IC}^{V} = \left\| \hat{Z}_F^{V} - Z_V^{\star} \right\|_1 . \tag{15}$$

The action term connects ODE-generated latents to executable actions; the visual term aligns the integrated prediction with compact future targets. These endpoint losses complement velocity supervision along the interpolated flow path.

G ACVB Token-Count Ablation

We compare one-, three-, and six-token ACVB variants using the same HIVE architecture and training procedure. Figure 13 shows cosine correspondences between compressed tokens and the 196 frozen DINOv3 patch tokens on four Franka tasks. With one token, the representation spreads over the entire image and does not separate the manipulated object from the background. Six tokens provide more capacity, but some tokens are redundant, while others mix the object and background or capture distracting regions. Three tokens provide the clearest separation of manipulation-relevant content with the smallest representation that preserves this structure.

Table 10 reports the corresponding 12-task Franka success rates. HIVE with three tokens has the highest mean success (57.08%), compared with 44.17% for one token and 48.25% for six tokens. Thus, the three-token setting is both task-effective and more compact than the six-token alternative.

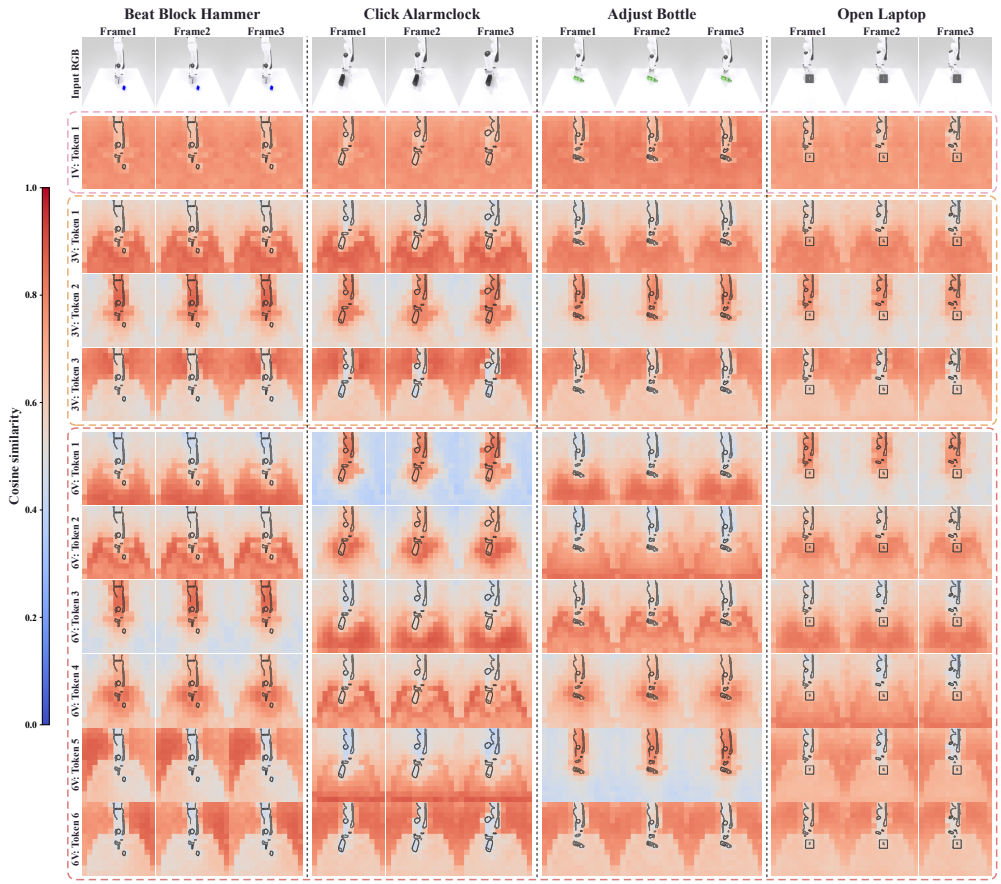


Figure 13: ACVB token-count ablation on four Franka tasks. Cosine correspondences between compressed tokens and the 196 DINOv3 patch tokens are shown for one, three, and six tokens. One token spreads attention across the image, whereas six tokens can be redundant or mix foreground and background; three tokens separate manipulation-relevant regions most clearly.

Table 10: Franka success rates (%) for different ACVB token counts. Means average the 12 third-person tasks.

Task	HIVE-1	HIVE-3	HIVE-6
Adjust Bottle	83	**93**	82
Beat Block Hammer	64	**75**	69
Click Alarmclock	86	**92**	79
Click Bell	77	**89**	86
Open Laptop	52	**61**	44
Place Shoe	8	**15**	9
Place Empty Cup	5	**23**	17
Move Playing Card Away	26	**47**	34
Press Stapler	45	**71**	65
Turn Switch	47	**51**	42
Rotate QR Code	22	**40**	26
Place Object Stand	15	**28**	26
Mean	44.17	**57.08**	48.25

H VISUAL LINEAR PROBE PROTOCOL

Data and interpretation. Table 2 covers 18 task–robot combinations: 12 Franka and six ARX-X5 tasks, with 1,150 trajectories and 150,433 raw frames. Each task has an independent probe fitted on all its valid in-domain windows and evaluated on those same windows. There is no validation/test split or OOD evaluation in this table. Metrics are averaged within each task and then equally over tasks; overlapping windows are not independent trajectories. The results quantify linear action fitting, not generalization.

Franka uses 50 trajectories per task (600 trajectories, 71,327 frames), third-person images, and 16-dimensional action labels. Its tasks are Adjust Bottle, Beat Block Hammer, Click Alarmclock, Click Bell, Move Playing Card Away, Open Microwave, Place Container Plate, Place Object Stand, Place Shoe, Press Stapler, Rotate QR Code, and Turn Switch. Relative to Table 1, this probe suite replaces Open Laptop and Place Empty Cup with Open Microwave and Place Container Plate. ARX-X5 uses head-camera images and 14-dimensional actions on Grab Roller, Handover Block, Move Can Pot, Place Burger Fries, Place Container Plate, and Stack Bowls Two. Each has 100 trajectories except Move Can Pot (50), totaling 550 trajectories and 79,106 frames. Place Container Plate is counted separately for the two robots, yielding 17 distinct task names.

Features and frozen weights. For each task, the largest-numbered available checkpoint in the specified action-reconstruction checkpoint directory supplies its configured EMA weights; probe metrics are not used for selection. DINOv3 and the trained task-specific compressor remain frozen. Images are converted to float32 in $[0, 1]$, resized bilinearly to 224×224, and normalized with ImageNet statistics. From the same DINOv3 pass, we extract either the 768-dimensional CLS feature or 196 patch tokens, excluding CLS and four register tokens. The compressor's last three output positions after LayerNorm provide the ACVB tokens. No flow network or action decoder is used.

ACVB-Mean averages the three tokens within each frame, retaining 768 dimensions like CLS. ACVB-Concat flattens them in token order to 2,304 dimensions. History features concatenate frames from oldest to newest, giving input dimensions 6,144 for CLS/Mean and 18,432 for Concat. Thus, CLS versus Mean controls probe capacity, whereas Concat has three times as many input weights. The compressor has received task supervision, unlike the frozen CLS representation; this is not a matched pretraining budget comparison.

Temporal labels and windows. Let q_t denote the raw recorded joint/gripper vector. Consistent with the policy preprocessing and this paper's timing, the action label is $a_t = q_{t+1}$. Single-frame probes map I_t to a_t, using $0 \le t \le T - 2$ for an episode of T records. History probes map $I_{t-7:t}$ to $A_{t:t+7} = [q_{t+1}, \ldots, q_{t+8}]$, using $7 \le t \le T - 9$. There are $T - 1$ and $\max(T - 15, 0)$ valid windows, respectively, constructed within each episode without boundary padding. All features within a setting use identical windows. The two settings have different sample sets and targets, so their raw errors do not isolate the benefit of history. Eight actions are predicted jointly in one linear mapping, with no autoregression, future images, robot state, historical actions, text, time index, or episode identifier as inputs.

Ridge fitting. For N samples, input dimension D, and output dimension $K = F_p d_a$ with probe horizon $F_p \in \{1, 8\}$, let $X \in \mathbb{R}^{N \times D}$ and $Y \in \mathbb{R}^{N \times K}$. Features use training-set means μ_j and population standard deviations s_j; values of $s_j < 10^{-6}$ are replaced by one:

$$Z_{ij} = \frac{X_{ij} - \mu_j}{s_j \sqrt{D}}, \qquad \min_{W,b} \frac{1}{N} \|ZW + \mathbf{1}b^\top - Y\|_F^2 + 0.01\|W\|_F^2. \tag{16}$$

The bias is unregularized. Labels remain in their original units and are centered only for solving, with their mean restored by the bias. The regularization strength is fixed for all tasks and settings without hyperparameter search. A float64 GPU direct solve uses the smaller primal or dual system with diagonal regularization $0.01N$ and relative residual below 10^{-8}. There are no optimizer, learning-rate, or epoch settings. Feature extraction uses batch size 32. The two reported settings require 108 probes in total (18 tasks, three features, two prediction settings).

Metrics. MSE and MAE average squared and absolute errors over all samples and output coordinates, including grippers and static-arm dimensions. With each output centered by its training mean

$\bar{Y}_k$, we use

$$R^2 = 1 - \frac{\sum_{i,k}(\hat{Y}_{ik} - Y_{ik})^2}{\sum_{i,k}(Y_{ik} - \bar{Y}_k)^2}. \tag{17}$$

This is variance-weighted multi-output R^2, not a simple average of per-joint scores. Absolute errors combine joint/gripper units and robot dimensions, so R^2 also provides a normalized fit measure. Normalized MSE equals $1 - R^2$ here, including after task macro-averaging, and is omitted as redundant. No cross-episode confidence intervals were computed.